$2.00

Reference Card For

MICROSOFT

EDITOR/ ASSEMBLER-PLUS

MICROSOFT
CONSUMER PRODUCTS

10800 Northeast Eighth, Suite 507
Bellevue, WA 98004
(206) 454-1315

Part No. 10E04
Printed in U.S.A.

RESET PROCEDURE

1. Press BREAK. If no response, proceed to step 2.
2. Press BREAK and RESET button on left rear of cpu if you have an expansion interface, otherwise press RESET button.
3. ENTER for "MEMORY SIZE?" (expansion interface case).
4. Enter "SYSTEM" for ">" prompt.
5. Enter one of three restart addresses:
   a. /17280 restarts and destroys contents of edit buffer.
   b. /17283 restarts with contents of edit buffer preserved.
   c. /17286 restarts with Z-BUG breakpoints and edit buffer preserved.

EDTASM-PLUS
SPECIAL CHARACTERS

Character	Description and Use
←	Backspace deletes the previous character on line input for Editor, Assembler, or Z-BUG.
Shift ←	Deletes the entire input line.
Shift ↑	Escape character, displayed as "$". Used in Editor and Z-BUG as delimiter. Cannot be deleted by backspace (←) or line delete (SHIFT ←) while in Z-BUG.
Shift@	Holds the display during multiple line listing in Editor, Assembler, or Z-BUG. Hitting any key except BREAK will restart the display.
Break	Generally used to return to "command" level in Editor, Assembler, or Z-BUG.
"C"	During "wait on error" wait in Assembler operation, C may be pressed to continue without error wait.

EDITOR RANGES OF LINES

Format	Example	Description
Start:End	100:300	Range of lines specified in this format is from "start" line number to "end" line number inclusive.
SLN!n	100!5	Range of lines specified in this format is from starting line number (SLN) and through the line count "n". Line count is number of lines, including the SLN, that is to be used in the range.
Offset	100 + 5 300 – 12	A positive or negative offset may be used to specify the "line n lines away" from a given line.

EDITOR COMMANDS

Command	Format	Description
Basic	*B	Return to Level II BASIC.
Copy	*CtIn,line1:line2,[inc]	Copy lines defined by range to area starting with line number "tIn" and with increment "inc".
Delete	*D[line1 [:line2]]	Delete range of lines specified.
Edit	*E[line1 [:line2]]	Block edit of range specified.
Find	*F[line1 [:line2]][$string]	Find string "string" and display all occurrences.
Insert	*I[line1 [,inc]]	Insert following lines starting at "line1" and incrementing by "inc".
Hard-copy	*H[line1 [:line2]]	Output lines defined by range to system printer.
Load	L[Ø filename]	Load file "filename" from cassette tape.
Move	*MtIn,line1:line2,[inc]	Move lines defined by range to area starting with line number "tIn" and with increment "inc".
Number	*N[line [,inc]]	Renumber lines starting at line "line" and incrementing by "inc".
Print	*P	Display the next 16 lines.
Print	*P[line1 [:line2]]	Display the range of lines specified.
Quash	*QA	Quash Assembler and Z-BUG to obtain more memory.
Quash	*QZ	Quash Z-BUG to obtain more memory.
Replace	R[line [,inc]]	Replace lines starting at "line" with following lines and increment "inc".
Substi-tute	*S[line1[:line2]][$string1] [$string2]	Substitute "string2" for every occurrence of "string1" over range.

Command	Format	Description
Type	*T[line1 [:line2]]	Print only text of lines in range defined.
Write	*W[bfilename]	Write out contents of edit buffer to cassette as file "filename".
E**X**tend	*X[line1 [:line2]]	Extend lines in insert mode over defined range.
↑ ↓		Scroll up (↑) or down (↓).

KEY:
1. line1:line2 format may be replaced by "for how many lines" format of sln!n where "sln" is starting line number and "n" is the line count. Line number offsets may also be used at any time.
2. Brackets indicate optional arguments.
3. "b" indicates optional blanks.
4. "line" defaults to current line if none entered.
5. "inc" defaults to last entered increment if none entered.

EDIT MODE SUBCOMMANDS

Subcommand	Format	Description
Again	A	Cancel all changes and restart.
Backspace	n←	Move cursor n positions left.
Change	nCstring	Change n characters at current cursor position to "string" characters.
Delete	nD	Delete n characters at current cursor position.
End edit	E	End edit and enter all changes without displaying remainder of line.
End edit	ENTER	End edit and enter all changes and display remainder of line.
E**X**tend line	X	Move to the end of line and enter insert mode.
Hack	H	Delete line from current cursor position to end and enter insert mode.
Insert	Istring	Enter insert mode before current cursor position. Terminate with SHIFT, up arrow↑.

Subcommand	Format	Description
Kill	nKs	Kill all characters from current cursor position to nth occurrence of "s".
List line	L	List line and position cursor to start of line in following line.
Quit	Q	Quit Edit mode and ignore all changes.
Search	nSs	Search for the nth occurrence of "s".
Space	nspace	Move cursor n positions right.

EDITOR ERROR MESSAGES

Message	Description and Corrective Action
BAD COMMAND	Editor does not recognize command. Is it valid?
BAD LINE NUMBER	Editor cannot find line number. Check to see if line is actually present in the edit buffer.
BAD PARAMETERS	Editor cannot decode the operands for the command. Check format.
BUFFER EMPTY	User has specified operation on a line when buffer is empty. Reload or reenter text.
BUFFER FULL	No more room in the edit buffer. Quash Z-BUG or Z-BUG and Assembler to get more space, or break up program into separate modules.
NEW LINE TOO LONG	Substitute has been performed that resulted in a line greater than 128 characters. Shorten the line.
NO ROOM BETWEEN LINES	Insert, move, or copy action resulted in too many lines to fit between existing lines. Renumber existing lines with larger increment.
STRING NOT FOUND	Find or Substitute command specified a string that cannot be found. Check the search string for proper characters.

ASSEMBLER PSEUDO-OPS

Pseudo-op	Format			Description
COND	—	COND	expression	Controls conditional assembly of following code until matching ENDC.
DEFB	(label)	DEFB	expression	Generates a single byte as defined by expression.
DEFL	label	DEFL	expression	Sets "label" equal to value of expression. "label" may be subsequently redefined by another DEFL.
DEFM	(label)	DEFM	'string'	Generates one byte ASCII characters for every character in "string".
DEFS	(label)	DEFS	expression	Reserves amount of storage in bytes equal to value of expression.
DEFW	(label)	DEFW	expression	Generates two bytes as defined by expression. Bytes are in standard Z-80 address format.
ENDC	—	ENDC	—	Terminates conditional assembly section started by matching COND.
ENDM	—	ENDM	—	Terminates macro definition started by MACRO.
END	—	END	(expression)	Denotes end of source code. Optional expression defines starting address for program.
EQU	label	EQU	expression	Equates "label" to value of expression.
MACRO	label	MACRO	(#P1,#P2, ... #PN)	Denotes start of macro definition terminated by ENDM. Optional parameter string defines dummy arguments.

Pseudo-op	Format			Description
ORG	—	ORG	expression	Defines origin of following code as defined by value of expression.

Key: "—" means not allowed. "( )" means optional. No parentheses means necessary label or expression.

ASSEMBLER/Z-BUG EXPRESSION EVALUATION

Operation	Symbol	Precedence	Notes
Addition	+	2	
Subtraction	–	2	
Multiplication	*	5	
Division	/(Assembler) .DIV. (Z-BUG)	5	"/" opens location in Z-BUG
Logical AND	.AND. or &	4	Takes two arguments
Logical OR	.OR. or !	3	Takes two arguments
Logical XOR	.XOR.	3	Takes two arguments
One's complement	.NOT.	6	Takes one argument
Shift	<	5	Positive count is left, negative count right
Modulo	.MOD.	5	Takes two arguments as in a divide
Equals	.EQU. or =	1	Takes two arguments
Not equals	.NEQ.	1	Takes two arguments
Parentheses	( )	7	Must never open an expression
Octal suffix	O or Q		
Hexadecimal suffix	H		
Decimal suffix	T or unsuffixed		
ASCII character	'char'		
Assembler location counter	$ or .		$ may be confused with Z-BUG escape character printout.

Key: 1 indicates lowest precedence; 7 indicates highest.

ASSEMBLER COMMANDS OTHER THAN PRINTING

To Assemble:

*Ab NAME b/SW/SW . . . /SW	Assembles from edit buffer. If object is output to cassette, Assembler uses optional name "NAME" (leading and trailing blanks are optional). /SW represents optional assembler switches (see "Assembler Switches").

To Define User Origin:

*O	Allows user to examine or define user origin for /MO assemblies.

To Quash Z-BUG:

*QZ	Quashes Z-BUG portion of EDTASM-PLUS, allowing additional area for edit buffer, symbol table, and in-memory object.

ASSEMBLER SWITCHES

Switch	Description
/AO	Assemble with absolute origin.
/IM	Assemble object directly into memory.
/LP	Output listing and symbol table to line printer.
/MO	Assemble with user-specifed origin.
/NL	Suppress listing printout or display.
/NO	Suppress object output.
/NS	Suppress symbol table printout or display.
/WE	Wait on errors until key depressed.

ASSEMBLER PRINTING COMMANDS

Command	Format	Description
LIST ON	*LIST ON	Assembler command. Enables display or printing of assembly listing until *LIST OFF.
LIST OFF	*LIST OFF	Assembler command. Disables display or printing of assembly listing until *LIST ON.
MLIST ON	*MLIST ON	Assembler command. Enables display or printing of macro expansions until *MLIST OFF.
MLIST OFF	*MLIST OFF	Assembler command. Disables display or printing of macro expansions until *MLIST ON.

ASSEMBLER SYMBOL TABLE CODES

Code	Description
D	Symbol has been defined by a DEFL pseudo-op.
F	Symbol is a macro name used erroneously before definition of the macro.
M	Symbol is a macro name.
R	Symbol has been erroneously defined more than one time.
U	Symbol has been erroneously referenced but never defined.

ASSEMBLER ERROR MESSAGES

Message	Description and Corrective Action
BAD ADDRESS	Invalid address defined for USRORG command or address above LAST or below USRORG on assembly. Use an address between FIRST and LAST.
BAD ADDRESSING MODE	Operands use incorrect addressing mode. Use valid addressing for the instruction.

Message	Description and Corrective Action
BAD EXPRESSION	Syntax of expression in source line incorrect. Redefine.
BAD MEMORY	Assemble into memory verify does not compare. Check RAM.
BAD LABEL	Invalid label has been used. Use 1 to 6 character label, starting with an alphabetic character. Do not use "reserved" (system) words. Check label.
BAD OPCODE	Invalid opcode or pseudo-op mnemonic has been used. Check spelling.
BRANCH OUT OF RANGE	Relative address displacement on JR or other instruction is greater than +127 or less than −128. Use JP or another instruction.
DIVISION BY 0	Expression used 0 as a divisor. Use non-zero value.
ENDC WITHOUT COND	ENDC encountered without a prior COND. Check for existence of COND.
ENDM WITHOUT MACRO	ENDM encountered without a prior MACRO. Check for existence of MACRO.
FIELD OVERFLOW	Instruction field cannot hold value used as an operand. Change value to fit.
MACRO FWD REF	Macro invoked before definition. Define macro before reference.
MISSING INFORMATION	Operands missing or incomplete in source line. Check format of instruction or pseudo-op.
MULTIPLY DEFINED SYMBOL	Reference to multiply defined symbol. (See next error message).
MULTIPLE DEFINITION	Same label was used again. Change to one unique label for each line.

Message	Description and Corrective Action
NESTED MACROS	Nested macros are not allowed. Redefine macros.
NO END STATEMENT	No END pseudo-op at end of source code. Add END.
STACK OVERFLOW	Expression too involved for stack. Simplify.
SYMBOL TABLE OVERFLOW	Too many labels used in source program. Shorten or delete labels (use $).
SYNTAX ERROR	Macro syntax incorrect. Check.
UNDEFINED SYMBOL	Symbol encountered that is not defined as label, EQU, DEFL, or MACRO. Define.

Z-BUG COMMANDS

Command	Format	Description
A	$A	Set ASCII type-out mode.
B	$B	Set byte examination mode.
C	(continue count)$C	Continue from current breakpoint for a number of times equal to continue count.
	$C	Resume program execution.
D	$D	Display current breakpoints.
E	$E	Return to system command mode.
G	(address)$G	Execute program starting at address "address".
	$G	Execute program pointed to by user's program counter.
I	(radix)$I	Set input radix to "radix". Radix may be 2 through 16.
L	⍽name$L	Load SYSTEM-format tape.
M	$M	Set mnemonic examination mode.

Command	Format	Description
N	$N	Set numeric debugging mode.
O	(radix)$O	Set output radix to "radix". Radix may be 8, 10, or 16.
P	(first)␣(last)␣(execution)␣NAME$P	Output SYSTEM-format tape from address "first" through address "last" with execution address "execution" and file name "NAME".
R	$R	Display all registers.
S	$S	Set symbolic debugging mode.
T	(first)␣(last)$T	Display block of locations from address "first" through address "last".
W	$W	Set word examination mode.
X	(address)$X	Set breakpoint at address "address".
Y	(value)$Y	Yank breakpoint number "value".
	$Y	Yank all breakpoints.
Z	*Z	(Editor/Assembler command.) Entry into Z-BUG.
↓	↓	Examine next location.
↑	↑	Examine previous byte.
/	/	Reopen current location.
=	(expression)=	Display value of expression.
;	;	Force one-time typeout of current location in numeric mode.
=	=	Force one-time type-out of current location in byte examination and numeric modes.

Command	Format	Description
→	→	Open new location based on current instruction (mnemonic mode) or current location address.
:	:	Force one-time typeout of current location in flags mode format.
@	(address)@	Single step from address "address".
	@	Single step from current instruction.

Key: $ is SHIFT↑ (ESCAPE)

Z-BUG ERROR MESSAGES

Message	Description
BAD EXPRESSION	Syntax of expression incorrect. Redefine.
BAD MEMORY	Memory verify does not compare. Check RAM.
DIVISION BY 0	Expression used 0 as a divisor. Use non-zero value.
STACK OVERFLOW	Expression too involved for stack. Simplify.
UNDEFINED SYMBOL	Symbol cannot be found in symbol table. Reenter or verify that it exists.
ZERR	One of the following: -Expression followed by ";". -Illegal ESCAPE command character. -Internal breakpoint problem. -Illegal input or output radix. -Attempt to set more than 8 breakpoints. -Attempt to set a breakpointed location. -Attempt to breakpoint register. -Attempt to breakpoint an RST instruction. -Continue count of 0. -Continue without breakpoint condition.